Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Please cancel claims 56, 57, 59, 69, 71, 72, and 76 without prejudice.

Please amend claims 25, 28, 58, 60-66, 68, 73, 78, and 79 as follows:

Claims 1-24 (canceled).

25. (currently amended): A method of condition generation comprising the steps of:

defining a set of arithmetic condition flags (ACFs);

~~determining side effects of a plurality of scalar conditions on an instruction by instruction basis;~~

~~setting a set of arithmetic scalar flags (ASFs) to save the determined side effects;~~

specifying a condition code within a first instruction ~~utilizing a compare instruction; and~~

executing the first instruction;

updating the ACFs based upon the specified condition code and a side effect resulting from the execution of the first instruction;

determining whether to execute a second instruction based on the state of the arithmetic condition flags; and

executing the second instruction if it is determined to execute the second instruction.

26. (original): The method of claim 25 further comprising the step of:

combining a previous state of the ACFs with the result of a condition code test specified by a current compare instruction to create a complex condition.

27. (original): The method of claim 25 wherein the condition code specifies a condition such as greater than (GT), less than (LT), equal (EQ) or less than or equal (LEQ).

28. (currently amended): The method of claim[27] 26 wherein the compare instruction is further utilized to specify the desired conditions to be tested and two source registers to be compared.

29. (original): The method of claim 28 wherein the compare instruction is further utilized to specify a data type covering packed data forms.

30. (original): The method of claim 28 wherein the compare instruction is further utilized to specify a Boolean combination specification field.

31. (original): The method of claim 26 further comprising the step of controlling branching in a sequence processor (SP) based upon the created complex condition.

32. (previously amended): The method of claim 26 further comprising the step of conditionally executing in a sequence processor (SP) and at least one processing element (PE) based on the created complex consisting of a Boolean combination of multiple conditions based upon the created complex condition.

33. (original): The method of claim 26 further comprising the step of conditionally executing on a combination of multiple conditions based upon the created complex condition.

Claims 34-55 (canceled).

56. (canceled)

57. (canceled)

58. (currently amended): The method of claim[57] 26 wherein the ~~at least one side effect~~ condition code is chosen from the group consisting of a carry indication, an overflow indication, a sign indication, and a zero indication.

59. (canceled)

60. (currently amended): ~~The method of claim 56~~ A method of conditional instruction execution comprising:

setting arithmetic scalar flags based on at least one side effect of the execution of a first instruction;

setting arithmetic condition flags based on the arithmetic scalar flags as specified by the first instruction;

determining whether to execute a second instruction based on the state of the arithmetic condition flags set by the first instruction; and

executing the second instruction if it is determined to execute the second instruction

wherein the second instruction comprises at least one selectable conditional execution instruction opcode bit specifying the conditional execution of the second instruction.

61. (currently amended): The method of claim[56] 60 wherein the arithmetic condition flags are set based on a Boolean combination of the arithmetic scalar flags.

62. (currently amended): The method of claim[56] 60 wherein the execution of the second instruction affects the arithmetic scalar flags.

63. (currently amended): The method of claim[56] 60 wherein the first instruction performs an operation on packed data comprising a plurality of data elements, and wherein the method further comprises:

setting one arithmetic scalar flag for each data element of the packed data.

64. (currently amended): The method of claim[56] 60 wherein the execution of the second instruction affects the state of the arithmetic condition flags.

65. (currently amended): The method of claim[56] 60 wherein the first instruction is executed by a first processing element and the second instruction is conditionally executed by a second processing element.

66. (currently amended): An apparatus for conditional instruction execution comprising:

~~means for setting arithmetic scalar flags based on at least one side effect of the execution of a first instruction;~~

executing a first instruction, the first instruction having one or more bits to indicate how to set arithmetic condition flags (ACFs);

means for setting arithmetic condition flags based on ~~the arithmetic scalar flags~~ said one or more bits and a side effect resulting from the execution of~~as specified by~~ the first instruction;

means for determining whether to execute a second instruction based on the state of the arithmetic condition flags set by the execution of the first instruction and said one or more bits; and

means for executing the second instruction if it is determined to execute the second instruction.

67. (original): The apparatus of claim 66 wherein the first instruction is a compare instruction.

68. (currently amended): The apparatus of claim 67 wherein ~~the at least one side effect~~ each of said one or more bits is chosen to indicate a condition from the group consisting of a carry indication, an overflow indication, a sign indication, and a zero indication.

69. (canceled)

70. (original): The apparatus of claim 66 wherein the second instruction comprises at least one selectable conditional execution instruction opcode bit specifying the conditional execution of the second instruction.

71. (canceled)

72. (canceled)

73. (currently amended): A processing apparatus for conditional instruction execution comprising:

a storage device for storing an arithmetic ~~scalar~~ condition flag;

an execution unit for executing a first instruction, and generating ~~an arithmetic scalar~~ condition state~~flag~~ as at least one side effect of the execution, ~~and storing the arithmetic scalar flag in the storage device~~; and

a generation unit for receiving the condition state, generating an arithmetic condition flag utilizing both the ~~stored arithmetic scalar flag~~ condition state and an opcode bit from the first instruction, storing the arithmetic condition flag in the storage device, said execution unit for conditionally executing a second instruction based on the state of the arithmetic condition flag.

74. (original): The processing apparatus of claim 73 wherein the first instruction is a compare instruction.

75. (original): The processing apparatus of claim 74 wherein the at least one side effect is chosen from the group consisting of a carry indication, an overflow indication, a sign indication, and a zero indication.

76. (canceled)

77. (original): The processing apparatus of claim 73 wherein the second instruction comprises at least one selectable conditional execution instruction opcode bit specifying the conditional execution of the second instruction.

78. (currently amended): The processing apparatus of claim 73 wherein the arithmetic condition flags are set based on a Boolean combination of the ~~arithmetic scalar flags~~ generated condition state.

79. (currently amended): The processing apparatus of claim 73 wherein the execution of the second instruction affects the ~~arithmetic scalar flags~~generated condition state.